Varying-Coefficient Models with Isotropic Gaussian Process Priors

Matthias Bussas MATTHIAS.BUSSAS.14@UCL.AC.UK
University College London, Department of Statistical Science
London WC1E 6BT, United Kingdom

Christoph Sawade CHRISTOPH@SOUNDCLOUD.COM
SoundCloud Ltd.
Greifswalder Str. 212-213, 10405 Berlin, Germany

Tobias Scheffer SCHEFFER@CS.UNI-POTSDAM.DE
University of Potsdam, Department of Computer Science
August-Bebel-Strasse 89, 14482 Potsdam, Germany

Niels Landwehr LANDWEHR@CS.UNI-POTSDAM.DE
University of Potsdam, Department of Computer Science
August-Bebel-Strasse 89, 14482 Potsdam, Germany

Editor:

Abstract

We study learning problems in which the conditional distribution of the output given the input varies as a function of additional task variables. In varying-coefficient models with Gaussian process priors, a Gaussian process generates the functional relationship between the task variables and the parameters of this conditional. Varying-coefficient models subsume hierarchical Bayesian multitask models, but also generalizations in which the conditional varies continuously, for instance, in time or space. However, Bayesian inference in varying-coefficient models is generally intractable. We show that inference for varying-coefficient models with isotropic Gaussian process priors resolves to standard inference for a Gaussian process that can be solved efficiently. MAP inference in this model resolves to multitask learning using task and instance kernels, and inference for hierarchical Bayesian multitask models can be carried out efficiently using graph-Laplacian kernels. We report on experiments for geospatial prediction.

1. Introduction

In standard settings of learning from independent and identically distributed *(iid)* data, labels y of training and test instances $\mathbf{x}$ are drawn independently and are governed by a fixed conditional distribution $p(y|\mathbf{x})$. A great variety of problem settings relax this assumption; they are widely referred to as *transfer learning*. We study a general transfer learning setting in which the conditional $p(y|\mathbf{x})$ is assumed to vary as a function of additional observable variables $\mathbf{t}$. The variables $\mathbf{t}$ can identify a specific domain that an observation was drawn from (as in *multitask learning*), or can be continuous attributes that describe, for instance, the time or location at which an observation was made (sometimes called *concept drift*).

A natural model for this setting is to assume a conditional $p(y|\mathbf{x};\mathbf{w})$ with parameters $\mathbf{w}$ that vary with $\mathbf{t}$. Such models are known as *varying-coefficient models* (*e.g.,* Hastie and Tibshirani, 1993; Gelfand et al., 2003). In *iid* learning, it is common to assume an isotropic Gaussian prior $p(\mathbf{w})$ over model parameters. When the parameters vary as a function of a task variable $\mathbf{t}$, it is natural to instead assume a Gaussian process (GP) prior over functions that map values of $\mathbf{t}$ to values of $\mathbf{w}$. A Gaussian process implements a prior $p(\boldsymbol{\omega})$ over functions $\boldsymbol{\omega} : \mathcal{T} \to \mathbb{R}^m$ that couple parameters $\mathbf{w} \in \mathbb{R}^m$ for different values of $\mathbf{t} \in \mathcal{T}$ and make it possible to generalize over different domains, time, or space. While this model allows to extend Bayesian inference naturally to a variety of transfer learning problems, inference in these varying-coefficient models for large problems is often impractical: It involves Kronecker products that result in matrices of size $nm \times nm$, with n the number of instances and m the number of attributes (Gelfand et al., 2003; Wheeler and Calder, 2006).

Alternatively, varying-coefficient models can be derived in a regularized risk minimization framework. Such models infer point estimates of parameters $\mathbf{w}$ for different observed values of $\mathbf{t}$ under some model that expresses how $\mathbf{w}$ changes smoothly with $\mathbf{t}$ (Fan and Zhang, 2008). At test time, point estimates of $\mathbf{w}$ are required for all $\mathbf{t}$ observed at the test data points. This is again computationally challenging because typically a separate optimization problem needs to be solved for each test instance. Most prominent are estimation techniques based on kernel-local smoothing (Fan and Zhang, 2008; Wu and Chiang, 2000; Fan and Huang, 2005).

In this paper, we explore Bayesian varying-coefficient models in conjunction with isotropic Gaussian process priors. An isotropic prior encodes the assumption that elements of the vector of model parameters are generated independently of one another; isotropic GP priors are in direct analogy to isotropic Gaussian priors that are widely used in *iid* learning. Our main theoretical result is that Bayesian inference in varying-coefficient models with isotropic Gaussian process priors is equal to Bayesian inference in a standard Gaussian process with a specific product kernel. The main practical implication of this result is that inference for varying-coefficient models becomes practical by using standard GP tools. Our theoretical result also leads to insights regarding existing transfer learning methods: First, we identify the exact modeling assumptions under which Bayesian inference amounts to multitask learning using a Gaussian process with task kernels and instance kernels (Bonilla et al., 2007). Secondly, we show that hierarchical Bayesian multitask models (*e.g.*, Gelman et al., 1995; Finkel and Manning, 2009) can be represented as Gaussian process priors; inference then resolves to inference in standard Gaussian processes with multitask kernels based on graph Laplacians (Evgeniou et al., 2005; Álvarez et al., 2011).

Our main empirical result is that varying-coefficient models with GP priors are an effective and efficient model for prediction problems in which the conditional distribution of the output given the input varies in time and geographical location. In our experiments, varying coefficient models outperform reference models for the problems of predicting rents and real-estate prices.

The paper is structured as follows. Section 2 describes the problem setting and the varying-coefficient model. Section 3 studies Bayesian inference and presents our main results. Section 4 presents experiments on prediction of real estate sales prices and monthly rents; Section 5 discusses related work and concludes.

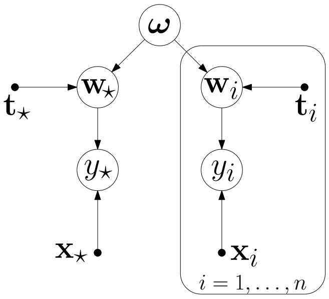

Figure 1: Generative process of the varying-coefficient model described in Section 2. Variables $\mathbf{x}_\star, y_\star, \mathbf{t}_\star, \mathbf{w}_\star$ denote the feature vector, label, task variable, and parameterization for a novel test instance.

2. Problem Setting and Model

This section defines a generative process which models a wide class of applications that are characterized by a conditional distribution $p(y|\mathbf{x}, \mathbf{w})$ whose parameterization $\mathbf{w}$ varies as a function of additional variables $\mathbf{t}$. Figure 1 shows a plate representation of the model.

A fixed set of instances $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^m$ is observable, along with values $\mathbf{t}_1, \ldots, \mathbf{t}_n \in \mathcal{T}$ of a *task variable*. The process starts by drawing a function $\boldsymbol{\omega} : \mathcal{T} \rightarrow \mathbb{R}^m$ according to a prior $p(\boldsymbol{\omega})$. The function $\boldsymbol{\omega}$ associates any task variable $\mathbf{t} \in \mathcal{T}$ with a corresponding parameter vector $\boldsymbol{\omega}(\mathbf{t}) \in \mathbb{R}^m$ that defines the conditional distribution $p(y|\mathbf{x}, \boldsymbol{\omega}(\mathbf{t}))$ for task $\mathbf{t} \in \mathcal{T}$. The domain $\mathcal{T}$ of the task variable depends on the application at hand. In the simplest case of *multitask learning*, $\mathcal{T} = \{1, \ldots, k\}$ is a set of task identifiers. In hierarchical Bayesian multitask models, a tree $\mathcal{G} = (\mathcal{T}, \mathbf{A})$ over the tasks $\mathcal{T} = \{1, \ldots, k\}$ reflects how tasks are related; we represent this tree by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$. We also study the setting of *concept drift* or *non-stationary learning* in which the conditional distribution of y given $\mathbf{x}$ varies smoothly in the task variables $\mathbf{t}$ that can, for instance, comprise time or space. In this case, $\mathcal{T} \subset \mathbb{R}^d$ is a continuous-valued space.

We model $p(\boldsymbol{\omega})$ using a zero-mean Gaussian process

$$\boldsymbol{\omega} \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\kappa}) \tag{1}$$

that generates vector-valued functions $\boldsymbol{\omega} : \mathcal{T} \rightarrow \mathbb{R}^m$. The process is specified by a matrix-valued kernel function $\boldsymbol{\kappa} : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}^{m \times m}$ that reflects closeness in $\mathcal{T}$. Here, $\boldsymbol{\kappa}(\mathbf{t}, \mathbf{t}') \in \mathbb{R}^{m \times m}$ is the matrix of covariances between components of the vectors $\boldsymbol{\omega}(\mathbf{t})$ and $\boldsymbol{\omega}(\mathbf{t}')$ for $\mathbf{t}, \mathbf{t}' \in \mathcal{T}$. We assume that the kernel function $\boldsymbol{\kappa}$ is isotropic; that is, $\boldsymbol{\kappa}(\mathbf{t}, \mathbf{t}') = k_{\mathcal{T}}(\mathbf{t}, \mathbf{t}')\mathbf{I}_{m \times m}$ for a positive semidefinite kernel function $k_{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$. This corresponds to the assumption that each dimension of the vector-valued function $\boldsymbol{\omega}$ is generated by an independent Gaussian process, and these Gaussian processes share a common kernel function $k_{\mathcal{T}}$. Note that this decoupling is not an independence assumption on attributes; it is instead analogous to the assumption of an isotropic normal prior for model parameters that justifies the standard ℓ_2-regularization. We use $\mathbf{K_T} \in \mathbb{R}^{n \times n}$ to denote the matrix given by evaluations $k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_j)$ of the kernel function $k_{\mathcal{T}}$. The process evaluates function $\boldsymbol{\omega}$

for all $\mathbf{t}_i$ to create parameter vectors $\mathbf{w}_1 = \boldsymbol{\omega}(\mathbf{t}_1), \ldots, \mathbf{w}_n = \boldsymbol{\omega}(\mathbf{t}_n)$. The process then concludes by generating labels y_i from an appropriate observation model,

$$y_i \sim p(y|\mathbf{x}_i, \mathbf{w}_i), \tag{2}$$

for instance, a standard linear model with Gaussian noise for regression or a logistic function of the inner product of $\mathbf{w}_i$ and $\mathbf{x}_i$ for classification.

The prediction problem is to infer the distribution of the label $y_\star$ for a new observation $\mathbf{x}_\star$ with task variable $\mathbf{t}_\star$. For notational convenience, we aggregate the training instances into matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ with row vectors $\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_n^\mathsf{T}$, the task variables into matrix $\mathbf{T} \in \mathbb{R}^{n \times d}$ with row vectors $\mathbf{t}_1^\mathsf{T}, \ldots, \mathbf{t}_n^\mathsf{T}$, the parameter vectors associated with training observations into a matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$ with row vectors $\mathbf{w}_1^\mathsf{T}, \ldots, \mathbf{w}_n^\mathsf{T}$, and the labels $y_1, \ldots, y_n$ into vector $\mathbf{y} \in \mathcal{Y}^n$.

In this model, the Gaussian process prior $p(\boldsymbol{\omega})$ over functions $\boldsymbol{\omega} : \mathcal{T} \to \mathbb{R}^m$ couples parameter vectors $\boldsymbol{\omega}(\mathbf{t})$ for different values $\mathbf{t}$ of the task variable. The hierarchical Bayesian model of multitask learning assumes a coupling of parameters based on a hierarchical Bayesian prior (*e.g.*, Gelman et al., 1995; Finkel and Manning, 2009). We will now show that the varying-coefficient model with isotropic GP prior subsumes hierarchical Bayesian multitask models by choice of an appropriate kernel function $\boldsymbol{\kappa}$ of the Gaussian process that defines $p(\boldsymbol{\omega})$. Together with results on inference presented in Section 3, this result shows how inference for hierarchical Bayesian multitask models can be carried out using a Gaussian process.

The following definition formalizes the hierarchical Bayesian multitask model.

Definition 1 (Hierarchical Bayesian Multitask Model) *Let $\mathcal{G} = (\mathcal{T}, \mathbf{A})$ denote a tree structure over a set of tasks $\mathcal{T} = \{1, \ldots, k\}$ given by an adjacency matrix $\mathbf{A}$, with $1 \in \mathcal{T}$ the root node. Let $\boldsymbol{\sigma} \in \mathbb{R}^k$ denote a vector with entries $\sigma_1, \ldots, \sigma_k$. The following process generates the distribution $p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \mathcal{G}, \boldsymbol{\sigma})$ over labels $\mathbf{y} \in \mathcal{Y}^n$ given instances $\mathbf{X}$, task variables $\mathbf{T}$, the task hierarchy $\mathcal{G}$, and variances $\boldsymbol{\sigma}$: The process first samples parameter vectors $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_k \in \mathbb{R}^m$ according to*

$$\bar{\mathbf{w}}_1 \sim \mathcal{N}(\bar{\mathbf{w}}|\mathbf{0}, \sigma_1^2 \mathbf{I}_{m \times m}) \tag{3}$$

$$\bar{\mathbf{w}}_l \sim \mathcal{N}(\bar{\mathbf{w}}|\bar{\mathbf{w}}_{pa(l)}, \sigma_l^2 \mathbf{I}_{m \times m}) \qquad 2 \leq l \leq k \tag{4}$$

where for $l \in \mathcal{T}$, $pa(l) \in \mathcal{T}$ is the unique node with $\mathbf{A}_{pa(l),l} = 1$; then, the process generates labels $y_i \sim p(y|\mathbf{x}_i, \bar{\mathbf{w}}_i)$, where $p(y|\mathbf{x}_i, \bar{\mathbf{w}}_i)$ is the same conditional distribution over labels given an instance and a parameter vector as was chosen for the varying-coefficient model in Equation 2. This process defines the hierarchical Bayesian multitask model.

The following proposition shows that the varying-coefficient model presented in Section 2 subsumes the hierarchical Bayesian multitask model.

Proposition 2 *Let $\mathcal{G} = (\mathcal{T}, \mathbf{A})$ denote a tree structure over a set of tasks $\mathcal{T} = \{1, \ldots, k\}$ given by an adjacency matrix $\mathbf{A}$. Let $\boldsymbol{\sigma} \in \mathbb{R}^k$ be a vector with entries $\sigma_1, \ldots, \sigma_k$. Let $k_{\mathbf{A},\boldsymbol{\sigma}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ be given by $k_{\mathbf{A},\boldsymbol{\sigma}}(t, t') = G_{t,t'}$, where $G_{i,j}$ denotes the entry at row i and column j of the matrix*

$$\mathbf{G} = (\mathbf{I}_{k \times k} - \mathbf{A})^{-1} \mathbf{S} \left(\mathbf{I}_{k \times k} - \mathbf{A}^\mathsf{T} \right)^{-1},$$

and $\mathbf{S} \in \mathbb{R}^{k \times k}$ denotes the diagonal matrix with entries $\sigma_1^2, \ldots, \sigma_k^2$. Let $\boldsymbol{\kappa} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{m \times m}$ be given by $\boldsymbol{\kappa}(t, t') = k_{\mathbf{A},\boldsymbol{\sigma}}(t, t') \mathbf{I}_{m \times m}$ and let $p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \boldsymbol{\kappa}) = \int p(\mathbf{y}|\mathbf{W}, \mathbf{X}) p(\mathbf{W}|\mathbf{T}; \boldsymbol{\kappa}) \mathrm{d}\mathbf{W}$ be the marginal distribution over labels given instances and task variables defined by the varying-coefficient model. Then it holds that $p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \boldsymbol{\kappa}) = p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \mathcal{G}, \boldsymbol{\sigma})$.

Proposition 2 implies that performing Bayesian prediction in the varying-coefficient model with the specified kernel function is identical to performing Bayesian inference in the hierarchical Bayesian multitask model. The proof is included in the appendix. In Proposition 2, entries $G_{t,t'}$ of $\mathbf{G}$ represent a task similarity derived from the tree structure $\mathcal{G}$. Instead of a tree structure over tasks, feature vectors describing individual tasks may also be given (Bonilla et al., 2007; Yan and Zhang, 2009). In this case, $\boldsymbol{\kappa}(t, t')$ can be computed from the task features; the varying-coefficient model then subsumes existing approaches for multitask learning with task features (see Section 3.3).

3. Inference

We now address the problem of inferring predictions $y_\star$ for instances $\mathbf{x}_\star$, and task variables $\mathbf{t}_\star$. Section 3.1 presents exact Bayesian solutions for regression; Section 3.2 discusses approximate Bayesian inference for classification. Section 3.3 derives existing multitask models as special cases.

3.1 Regression

This subsection studies linear regression models of the form $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|\mathbf{x}^\mathsf{T}\mathbf{w}, \tau^2)$. Note that by substituting for the slightly heavier notation $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|\Phi(\mathbf{x})^\mathsf{T}\mathbf{w}, \tau^2)$, this treatment also covers finite-dimensional feature maps. The predictive distribution for test instance $\mathbf{x}_\star$ with task variable $\mathbf{t}_\star$ is obtained by integrating over the possible parameter values $\mathbf{w}_\star$ of the conditional distribution that has generated value $y_\star$:

$$p(y_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) = \int p(y_\star|\mathbf{x}_\star, \mathbf{w}_\star) p(\mathbf{w}_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{t}_\star) \mathrm{d}\mathbf{w}_\star, \tag{5}$$

where the posterior over $\mathbf{w}_\star$ is obtained by integrating over the joint parameter values $\mathbf{W}$ that have generated the labels $\mathbf{y}$ for instances $\mathbf{X}$ and task variables $\mathbf{T}$:

$$p(\mathbf{w}_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{t}_\star) = \int p(\mathbf{w}_\star|\mathbf{W}, \mathbf{T}, \mathbf{t}_\star) p(\mathbf{W}|\mathbf{X}, \mathbf{y}, \mathbf{T}) \mathrm{d}\mathbf{W}. \tag{6}$$

Posterior distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{y}, \mathbf{T})$ in Equation 6 depends on the likelihood function—the linear model—and the GP prior $p(\boldsymbol{\omega})$. The extrapolated posterior $p(\mathbf{w}_\star|\mathbf{W}, \mathbf{T}, \mathbf{t}_\star)$ for test instance $\mathbf{x}_\star$ with task variable $\mathbf{t}_\star$ depends on the Gaussian process. The following theorem states how the predictive distribution given by Equation 5 can be computed.

Theorem 3 (Bayesian Predictive Distribution) *Let $\mathcal{Y} = \mathbb{R}$, $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|\mathbf{x}^\mathsf{T}\mathbf{w}, \tau^2)$, and let the kernel matrix $\mathbf{K_T}$ be positive definite. Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be a matrix with components $k_{ij} = \mathbf{x}_i^\mathsf{T}\mathbf{x}_j k_\mathcal{T}(\mathbf{t}_i, \mathbf{t}_j)$ and $\mathbf{k} \in \mathbb{R}^n$ be a vector with components $k_i = \mathbf{x}_i^\mathsf{T}\mathbf{x}_\star k_\mathcal{T}(\mathbf{t}_i, \mathbf{t}_\star)$. Then, the predictive distribution for the varying-coefficient model defined in Section 2 is given by*

$$p(y_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) = \mathcal{N}(y_\star|\mu, \sigma^2 + \tau^2) \tag{7}$$

with

$$\begin{aligned} \mu &= \mathbf{k}^\mathsf{T}(\mathbf{K} + \tau^2\mathbf{I}_{n\times n})^{-1}\mathbf{y}, \\ \sigma^2 &= \mathbf{x}_\star^\mathsf{T}\mathbf{x}_\star k_\mathcal{T}(\mathbf{t}_\star, \mathbf{t}_\star) - \mathbf{k}^\mathsf{T}(\mathbf{K} + \tau^2\mathbf{I}_{n\times n})^{-1}\mathbf{k}. \end{aligned}$$

Before we prove Theorem 3, we highlight three observations about this result. First, the distribution $p(y_\star|\mathbf{X},\mathbf{y},\mathbf{T},\mathbf{x}_\star,\mathbf{t}_\star)$ has a surprisingly simple form. It is identical to the predictive distribution of a standard Gaussian process that uses concatenated vectors $(\mathbf{x}_1,\mathbf{t}_1),\ldots,(\mathbf{x}_n,\mathbf{t}_n)\in\mathcal{X}\times\mathcal{T}$ as training instances, labels $y_1,\ldots,y_n$, and the product kernel function $k((\mathbf{x}_i,\mathbf{t}_i),(\mathbf{x}_j,\mathbf{t}_j))=\mathbf{x}_i^\mathsf{T}\mathbf{x}_j k_\mathcal{T}(\mathbf{t}_i,\mathbf{t}_j)$.

Secondly, instances $\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x}_\star\in\mathcal{X}$ only enter Equation 7 in the form of inner products. The model can therefore directly be kernelized by defining the kernel matrix as $\mathbf{K}_{ij}=k_\mathcal{X}(\mathbf{x}_i,\mathbf{x}_j)k_\mathcal{T}(\mathbf{t}_i,\mathbf{t}_j)$ with kernel function $k_\mathcal{X}(\mathbf{x}_i,\mathbf{x}_j)=\Phi(\mathbf{x}_i)^\mathsf{T}\Phi(\mathbf{x}_j)$ where Φ maps to a reproducing kernel Hilbert space. When the feature space is finite, then $\boldsymbol{\omega}$ maps the $\mathbf{t}_i$ to a finite-dimensional $\mathbf{w}_i$ and Theorem 3 implies a Bayesian predictive distribution derived from the generative process that Section 2 specifies. When the reproducing kernel Hilbert space does not have a finite dimension, Section 2 does no longer specify a corresponding proper generative process because $p(\mathbf{w}_1,\ldots,\mathbf{w}_n|\mathbf{T})$ would otherwise become infinite-dimensionally normally distributed. However, given the finite sample $\mathbf{X}$ and $\mathbf{T}$, a Mercer map (see, *e.g.,* Schölkopf and Smola, 2002, Section 2.2.4) constitutes a finite-dimensional space $\mathbb{R}^n$ for which Section 2 again characterizes a corresponding generative process.

Thirdly and finally, Theorem 3 shows how Bayesian inference in varying-coefficient models with isotropic priors can be implemented much more efficiently than in general varying-coefficient models. Bayesian inference in varying-coefficient models in the parameter space generally involves matrices of size $nm\times nm$ because it needs to take the overall covariance structure into account; the algorithm of Gelfand et al. infers the covariance matrix under an inverse Wishart prior using a sliced Gibbs sampler over parameter values Gelfand et al. (2003). This makes inference impractical for large-scale problems. Theorem 3 shows that under the isotropy assumption, the latent parameter vectors $\mathbf{w}_1,\ldots,\mathbf{w}_n$ can be integrated out, which results in a GP formulation in which the covariance structure over parameter vectors resolves to an $n\times n$ product-kernel matrix.

Proof of Theorem 3. Let w_{ir} and $w_{\star r}$ denote the r-th elements of vectors $\mathbf{w}_i$ and $\mathbf{w}_\star$, and let x_{ir} and $x_{\star r}$ denote the r-th elements of vectors $\mathbf{x}_i$ and $\mathbf{x}_\star$. Let $\mathbf{z}_\star=(z_1,\ldots,z_n,z_\star)^\mathsf{T}\in\mathbb{R}^{n+1}$ with $z_i=\mathbf{x}_i^\mathsf{T}\mathbf{w}_i$ and $z_\star=\mathbf{x}_\star^\mathsf{T}\mathbf{w}_\star$. Because $\mathbf{w}_1,\ldots,\mathbf{w}_n,\mathbf{w}_\star$ are evaluations of the function $\boldsymbol{\omega}$ drawn from a Gaussian process (Equation 1), they are jointly Gaussian distributed and thus $z_1,\ldots,z_n,z_\star$ are also jointly Gaussian (*e.g.,* Murphy, 2012, Chapter 10.2.5). Because $\boldsymbol{\omega}$ is drawn from a zero-mean process, it holds that $\mathbb{E}[z_i]=\mathbb{E}[\sum_{r=1}^m x_{ir}w_{ir}]=\sum_{r=1}^m x_{ir}\mathbb{E}[w_{ir}]=0$ as well as $\mathbb{E}[z_\star]=0$ and therefore

$$p(\mathbf{z}_\star|\mathbf{X},\mathbf{T},\mathbf{x}_\star,\mathbf{t}_\star)=\mathcal{N}(\mathbf{z}_\star|\mathbf{0},\mathbf{C})$$

where $\mathbf{C}\in\mathbb{R}^{(n+1)\times(n+1)}$ denotes the covariance matrix. For the covariances $\mathbb{E}[z_i z_j]$ it holds that

$$\begin{aligned}
\mathbb{E}\left[z_i z_j\right] &= \mathbb{E}\left[\mathbf{x}_i^\mathsf{T}\mathbf{w}_i\mathbf{x}_j^\mathsf{T}\mathbf{w}_j\right]\\
&= \mathbb{E}\left[\left(\sum_{s=1}^m x_{is}w_{is}\right)\left(\sum_{r=1}^m x_{jr}w_{jr}\right)\right]\\
&= \sum_{s=1}^m\sum_{r=1}^m x_{is}x_{jr}\mathbb{E}\left[w_{is}w_{jr}\right]\\
&= \sum_{s=1}^m x_{is}x_{js}\mathbb{E}\left[w_{is}w_{js}\right] \qquad (8)\\
&= \mathbf{x}_i^\mathsf{T}\mathbf{x}_j k_\mathcal{T}(\mathbf{t}_i,\mathbf{t}_j). \qquad (9)
\end{aligned}$$

In Equations 8 and 9 we exploit the isotropy of the Gaussian process prior: the covariance $\mathbb{E}[w_{is}w_{jr}]$ is the element in row s and column r of the matrix $\boldsymbol{\kappa}(\mathbf{t}_i, \mathbf{t}_j) \in \mathbb{R}^{m \times m}$ obtained by evaluating the kernel function $\boldsymbol{\kappa} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{m \times m}$ at $(\mathbf{t}_i, \mathbf{t}_j)$; the isotropy assumption $\boldsymbol{\kappa}(\mathbf{t}, \mathbf{t}') = k_{\mathcal{T}}(\mathbf{t}, \mathbf{t}')\mathbf{I}_{m \times m}$ means that this matrix is diagonal with $\mathbb{E}[w_{is}w_{jr}] = 0$ for $s \neq r$ and $\mathbb{E}[w_{is}w_{js}] = k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_j)$ (see Section 2). We analogously derive

$$\mathbb{E}[z_i z_\star] = \mathbf{x}_i^\mathsf{T} \mathbf{x}_\star k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_\star), \tag{10}$$

$$\mathbb{E}[z_\star z_\star] = \mathbf{x}_\star^\mathsf{T} \mathbf{x}_\star k_{\mathcal{T}}(\mathbf{t}_\star, \mathbf{t}_\star). \tag{11}$$

Equations 9, 10 and 11 define the covariance matrix $\mathbf{C}$, yielding

$$p(\mathbf{z}_\star | \mathbf{X}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) = \mathcal{N}\left(\mathbf{z}_\star | \mathbf{0}, \begin{pmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^\mathsf{T} & k_\star \end{pmatrix}\right)$$

where $k_\star = \mathbf{x}_\star^\mathsf{T} \mathbf{x}_\star k_{\mathcal{T}}(\mathbf{t}_\star, \mathbf{t}_\star)$. For $\mathbf{y}_\star = (y_1, \ldots, y_n, y_\star)$ it now follows that

$$p(\mathbf{y}_\star | \mathbf{X}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) = \mathcal{N}\left(\mathbf{y}_\star | \mathbf{0}, \begin{pmatrix} \mathbf{K} + \tau^2 \mathbf{I}_{n \times n} & \mathbf{k} \\ \mathbf{k}^\mathsf{T} & k_\star + \tau^2 \end{pmatrix}\right). \tag{12}$$

The claim now follows by applying standard Gaussian identities to compute the conditional distribution $p(y_\star | \mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star)$ from Equation 12. ■

3.2 Classification

The result given by Theorem 3 can be extended to classification settings with $\mathcal{Y} = \{0, 1\}$ by using non-Gaussian likelihoods $p(y|z)$ that generate labels $y \in \mathcal{Y}$ given outputs $z \in \mathbb{R}$ of the linear model.

Theorem 4 (Bayesian predictive distribution for non-Gaussian likelihoods) *Let* $\mathcal{Y} = \{0, 1\}$. *Let* $p(y_i | \mathbf{x}_i, \mathbf{w}_i)$ *be given by a generalized linear model, defined by* $z_i \sim \mathcal{N}(z | \mathbf{w}_i^\mathsf{T} \mathbf{x}_i, \tau^2)$ *and* $y_i \sim p(y|z_i)$. *Let* $p(y_\star | \mathbf{x}_\star, \mathbf{w}_\star)$ *be given by* $z_\star \sim \mathcal{N}(z | \mathbf{w}_\star^\mathsf{T} \mathbf{x}_\star, \tau^2)$ *and* $y_\star \sim p(y|z_\star)$. *Let furthermore* $\mathbf{z} = (z_1, \ldots, z_n)^\mathsf{T} \in \mathbb{R}^n$.

Let the kernel matrix $\mathbf{K}_\mathbf{T}$ *be positive definite, and let* $\mathbf{K} \in \mathbb{R}^{n \times n}$ *be a matrix with components* $k_{ij} = \mathbf{x}_i^\mathsf{T} \mathbf{x}_j k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_j)$ *and* $\mathbf{k} \in \mathbb{R}^n$ *a vector with components* $k_i = \mathbf{x}_i^\mathsf{T} \mathbf{x}_\star k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_\star)$. *Then, the predictive distribution for the GP model defined in Section 2 is given by*

$$p(y_\star | \mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) \propto \iint p(y_\star | z_\star) \mathcal{N}(z_\star | \mu_\mathbf{z}, \sigma_\mathbf{z}^2) p(\mathbf{y}|\mathbf{z}) \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{K} + \tau^2 \mathbf{I}_{n \times n}) \mathrm{d}\mathbf{z} \mathrm{d}z_\star \tag{13}$$

with

$$\mu_\mathbf{z} = \mathbf{k}^\mathsf{T} (\mathbf{K} + \tau^2 \mathbf{I}_{n \times n})^{-1} \mathbf{z},$$
$$\sigma_\mathbf{z}^2 = \mathbf{x}_\star^\mathsf{T} \mathbf{x}_\star k_{\mathcal{T}}(\mathbf{t}_\star, \mathbf{t}_\star) - \mathbf{k}^\mathsf{T} (\mathbf{K} + \tau^2 \mathbf{I}_{n \times n})^{-1} \mathbf{k} + \tau^2.$$

A straightforward calculation shows that Equation 13 is identical to the predictive distribution of a standard Gaussian process that uses concatenated vectors $(\mathbf{x}_1, \mathbf{t}_1), \ldots, (\mathbf{x}_n, \mathbf{t}_n) \in \mathcal{X} \times \mathcal{T}$ as training instances, labels $y_1, \ldots, y_n$, the product kernel $k((\mathbf{x}_i, \mathbf{t}_i), (\mathbf{x}_j, \mathbf{t}_j)) = \mathbf{x}_i^\mathsf{T} \mathbf{x}_j k_{\mathcal{T}}(\mathbf{t}_i, \mathbf{t}_j)$, and likelihood function $p(y|z)$. For non-Gaussian likelihoods, exact inference in Gaussian processes is generally intractable, but approximate inference methods based on, *e.g.*, Laplace approximation, variational inference or expectation propagation are available.

Proof of Theorem 4. Rewriting $p(y_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star)$ in terms of a marginalization over the variables $\mathbf{z}$ and $z_\star$ leads to:

$$
\begin{aligned}
p(y_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) &= \int p(y_\star|z_\star)p(z_\star|\mathbf{X}, \mathbf{y}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star)\mathrm{d}z_\star \\
&= \iint p(y_\star|z_\star)p(z_\star|\mathbf{X}, \mathbf{z}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star)p(\mathbf{z}|\mathbf{X}, \mathbf{y}, \mathbf{T})\mathrm{d}\mathbf{z}\mathrm{d}z_\star \\
&\propto \iint p(y_\star|z_\star)p(z_\star|\mathbf{X}, \mathbf{z}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star)p(\mathbf{y}|\mathbf{z})p(\mathbf{z}|\mathbf{X}, \mathbf{T})\mathrm{d}\mathbf{z}\mathrm{d}z_\star.
\end{aligned}
$$

The proof now quickly follows from Theorem 3 and derivations in the proof of Theorem 3: Equation 7 implies $p(z_\star|\mathbf{X}, \mathbf{z}, \mathbf{T}, \mathbf{x}_\star, \mathbf{t}_\star) = \mathcal{N}(z_\star|\mu_\mathbf{z}, \sigma^2_\mathbf{z})$, Equation 12 implies $p(\mathbf{z}|\mathbf{X}, \mathbf{T}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{K} + \tau^2\mathbf{I}_{n\times n})$. ■

3.3 Product Kernels in Transfer Learning

Sections 3.1 and 3.2 have shown that inference in the varying-coefficient model is equivalent to inference in standard Gaussian processes with products of task kernels and instance kernels. Similar product kernels are used in several existing transfer learning models. Our results identify the generative assumptions that underlie these models by showing that the product kernels which they employ can be derived from the assumption of a varying-coefficient model with isotropic GP prior and an appropriate kernel function.

Bonilla et al. (2007) study a setting in which there is a discrete set of k tasks, which are described by task-specific attribute vectors $\mathbf{t}_1, \ldots, \mathbf{t}_k$. They study a Gaussian process model based on concatenated feature vectors $(\mathbf{x}, \mathbf{t})$ and a product kernel $k((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')) = k_\mathcal{X}(\mathbf{x}, \mathbf{x}')k_\mathcal{T}(\mathbf{t}, \mathbf{t}')$, where $k_\mathcal{X}(\mathbf{x}, \mathbf{x}')$ reflects instance similarity and $k_\mathcal{T}(\mathbf{t}, \mathbf{t}')$ reflects task similarity. Theorems 3 and 4 identify the generative assumptions underlying this model: a varying-coefficient model with isotropic Gaussian process prior and kernel $k_\mathcal{T}$ generates task-specific parameter vectors in a reproducing Hilbert space of the instance kernel $k_\mathcal{X}$; a linear model in that Hilbert space generates the observed labels.

Evgeniou et al. (2005) and Álvarez et al. (2011) study multitask-learning problems in which task similarities are given in terms of a task graph. Their method uses the product of an instance kernel and the graph-Laplacian kernel of the task graph. We will now show that, when the task graph is a tree, that kernel emerges from Proposition 2. This signifies that, when the task graph is a tree, the graph regularization method of Evgeniou et al. (2005) is the dual formulation of hierarchical Bayesian multitask learning, and therefore Bayesian inference for hierarchical Bayesian models can be carried out efficiently using a standard Gaussian process with a graph-Laplacian kernel.

Definition 5 (Graph-Laplacian Multitask Kernel) *Let $\mathcal{G} = (\mathcal{T}, \mathbf{M})$ denote a weighted undirected graph structure over a set of tasks $\mathcal{T} = \{1, \ldots, k\}$ given by a symmetric adjacency matrix $\mathbf{M} \in \mathbb{R}^{k\times k}$, where $\mathbf{M}_{i,j}$ defines the positive weight of the edge between tasks i and j or $\mathbf{M}_{i,j} = 0$ if no such edge exists. Let $\mathbf{D}$ denote the weighted degree matrix of the graph, and $\mathbf{L} = \mathbf{D} + \mathbf{R} - \mathbf{M}$ the* *graph Laplacian, where a diagonal matrix* $\mathbf{R}$ *that acts as a regularizer has been added to the degree matrix (Álvarez et al., 2011). The kernel function* $k_{\mathbf{M},\mathbf{R}} : (\mathcal{X} \times \mathcal{T}) \times (\mathcal{X} \times \mathcal{T}) \to \mathbb{R}$ *given by*

$$k_{\mathbf{M},\mathbf{R}}((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')) = \mathbf{L}^{\dagger}_{\mathbf{t},\mathbf{t}'} \mathbf{x}^{\mathsf{T}} \mathbf{x}',$$

where $\mathbf{L}^{\dagger}$ *is the pseudoinverse of* $\mathbf{L}$*, will be referred to as the* graph-Laplacian multitask kernel.

The following proposition states that the graph-Laplacian multitask kernel is equal to the kernel that emerges in the dual formulation of hierarchical Bayesian multitask learning (Definition 1).

Proposition 6 *Let* $\mathcal{G} = (\mathcal{T}, \mathbf{A})$ *denote a directed tree structure given by an adjacency matrix* $\mathbf{A}$*. Let* $\boldsymbol{\sigma} \in \mathbb{R}^k$ *be a vector with entries* $\sigma_1, \ldots, \sigma_k$*. Let* $\mathbf{B} \in \mathbb{R}^{k \times k}$ *denote the diagonal matrix with entries* $0, \sigma_2^{-2}, \ldots, \sigma_k^{-2}$*, let* $\mathbf{R} \in \mathbb{R}^{k \times k}$ *denote the diagonal matrix with entries* $\sigma_1^{-2}, 0, \ldots, 0$*, let* $\mathbf{M} = \mathbf{BA} + (\mathbf{BA})^{\mathsf{T}}$*, and let* $k_{\mathbf{A},\boldsymbol{\sigma}}(\mathbf{t}, \mathbf{t}')$ *be defined as in Proposition 2. Then*

$$k_{\mathbf{M},\mathbf{R}}((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')) = k_{\mathbf{A},\boldsymbol{\sigma}}(\mathbf{t}, \mathbf{t}') \mathbf{x}^{\mathsf{T}} \mathbf{x}'.$$

Note that in Proposition 6, $\mathbf{BA}$ is an adjacency matrix in which an edge from node i to node j is weighted by the respective precision σ_j^{-2} of the conditional distribution (Equation 4); adding the transpose yields a symmetric matrix $\mathbf{M}$ of task relationship weights. The precision σ_1^{-2} of the root node prior is subsumed in the regularizer $\mathbf{R}$. The proof is included in the appendix.

4. Empirical Study

In this section, we study the efficiency and accuracy of different varying-coefficient models and baselines for geospatial and temporal regression and classification problems. We focus on the problems of predicting real estate prices and monthly housing rents.

For real estate price prediction, we acquire records of real-estate sales in New York City for sales dating from January 2003 to December 2009 in June 2013 through the NYC Open Data initiative[1] . Input variables include the floor space, plot area, property class (such as family home, residential condominium, office, or store), date of construction of the building, and the number of residential and commercial units in the building. After binarization of multi-valued attributes there are 94 numeric attributes in the data set. For regression, the sales price serves as target variable y; we also study a classification problem in which y is a binary indicator that distinguishes between transactions with a price above the median of 450,000 dollars from transactions below it. Date and address for every sale are available; we transform addresses into geographical latitude and longitude using an inverse geocoding service based on OpenStreetMap data. We encode the sales date and geographical latitude and longitude of the property as task variable $\mathbf{t} \in \mathbb{R}^3$.

Price and attributes in sales records vary widely; for instance, prices range from one dollar to four billion dollars, and the floor space from one square foot to fourteen million square feet. A substantial number of records contain either errors or document transactions in which the valuations do not reflect the actual market values: for instance, Manhattan condominiums that sold for one dollar, and one-square-foot lots that sold for massive prices. In order to filter most off-market transactions by means of a simple policy, we only include records of sales within a price range of 100,000 to 1,000,000 dollars, a property area range of 500 to 5,000 square feet, and a land area

[1] https://nycopendata.socrata.com/.

range of 500 to 10,000 square feet. Approximately 80% of all records fall into these brackets. Additionally, we remove all records with missing values. After preprocessing, the data set contains 231,708 sales records. We divide the records, which span dates from January 2003 to December 2009, into 25 consecutive blocks. Models are trained on a set of n instances sampled randomly from a window of five blocks of historical data and evaluated on the subsequent block; results are averaged over all blocks.

For rent prediction, we acquire records on the monthly rent paid for privately rented apartments and houses in the states of California and New York from the 2013 American Community Survey's ASC public use microdata sample files[2]. Input variables include the number of rooms, number of bedrooms, the duration for which the contract has been running, the construction year of the building, the type of building (mobile home, trailer, or boat; attached or detached family house; apartment building), and variables that describe technical facilities (*e.g.,* variables related to internet access, type of plumbing, and type of heating). After binarization of multi-valued attributes there are 24 numerical attributes in the data. We study a regression problem in which the target variable y is the monthly rent, and a classification problem in which y is a binary indicator that distinguishes contracts with a monthly rent above the median of 1,200 dollars from those with a rent below the median. For each record, the geographical location is available in the form of a public use microdata area (PUMA) code[3]. We translate PUMA codes to geographical latitude and longitude by associating each record with the longitude-latitude-centroid of the corresponding public use microdata area; these geographical latitudes and longitudes constitute the task variable $\mathbf{t} \in \mathbb{R}^2$. We remove all records with missing values. The preprocessed data sets contain 36,785 records (state of California) and 17,944 records (state of New York). Models are evaluated using 20-fold cross validation; in each fold, a random subset of n training instances is sampled randomly from the respective training fold.

We study the varying-coefficient model with isotropic GP prior introduced in Section 2 with a Matérn kernel $k_{\mathcal{T}}(\mathbf{t}, \mathbf{t}')$. Predictions are obtained from Theorem 3, using either a linear or also a Matérn kernel function $k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}')$ (denoted by *isoVCM*^{lin} and *isoVCM*^{mat}, respectively). We compare with the varying-coefficient model with nonisotropic GP prior by Gelfand et al. (2003), in which the covariances are inferred from data (denoted by *Gelfand*). Furthermore, we compare with the kernel-local smoothing varying-coefficient model of Fan and Zhang (2008) that infers point estimates of model parameters. We study this model using a linear feature map (*Fan & Zhang*^{lin}) and a nonlinear feature map constructed from a Matérn kernel (*Fan & Zhang*^{mat}). Fan and Zhang (2008) do not regularize parameter estimates in their original model, we added an ℓ_2-regularizer as this improved predictive performance.

We finally compare against an *iid* baseline that assumes that $p(y|\mathbf{x})$ is constant in $\mathbf{t}$, implemented by a standard Gaussian process with a linear ($GP^{lin}_{\mathbf{x}}$) or Matérn ($GP^{mat}_{\mathbf{x}}$) kernel, and with a standard Gaussian process that simply concatenates instance and task attribute vectors into vectors $(\mathbf{x}, \mathbf{t})$ (denoted $GP^{lin}_{\mathbf{x},\mathbf{t}}$ and $GP^{mat}_{\mathbf{x},\mathbf{t}}$).

For classification, we use logistic likelihood functions in our model (Theorem 4), and also in the GP baselines and the kernel-local smoothing varying-coefficient model of Fan and Zhang (2008). All kernel parameters, as well as the observation noise parameter τ of Theorem 3 and the observation noise parameters of the standard GP models are tuned according to marginal likelihood on

[2] http://factfinder.census.gov/faces/affhelp/jsf/pages/metadata.xhtml?lang=en&type=document&id=document.en.ACS_pums_csv_2013#main_content.

[3] https://www.census.gov/geo/reference/puma.html.

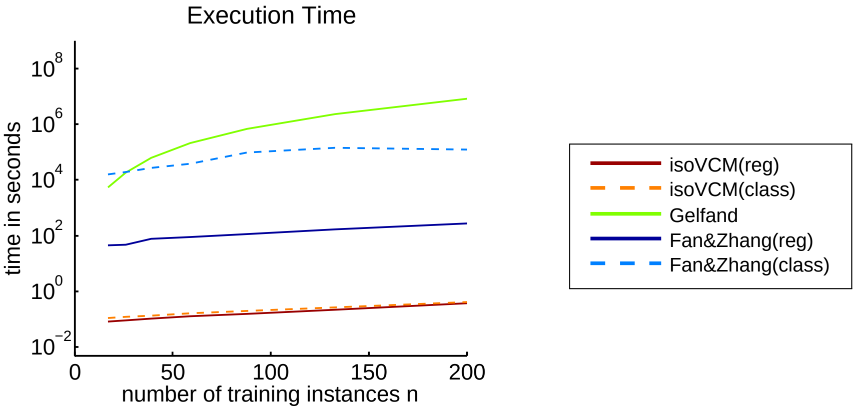

Figure 2: Execution time of *isoVCM* and reference methods over training set size n.

the training data. The regularization parameter of the kernel-local smoothing varying-coefficient model and its kernel parameter h (see Fan and Zhang, 2008) are tuned on the training data by cross-validation. The *isoVCM* model and all GP baselines are implemented based on the GPML Gaussian process toolbox (Rasmussen and Nickisch, 2010). Inference is carried out using the FITC approximation based on a low-rank approximation to the exact covariance matrix with 1,000 randomly sampled inducing points (Snelson and Ghahramani, 2005), and using Laplace approximation for classification.

First, we compare the execution time of the GP inference that results from Theorem 3 with the execution time of the primal inference procedure of Gelfand et al. (2003) and the execution time of the kernel-local smoothing varying-coefficient model of Fan and Zhang (2008). Figure 2 shows the execution time for model training and prediction on one block of test instances in the real estate price prediction task as a function of the training set size n (CPU core seconds, Intel Xeon 5520, 2.26 GHz). For the model of Gelfand et al., the most expensive step during inference is computation of the inverse of a Cholesky decomposition of an $nm \times nm$ matrix, which needs to be performed within each Gibbs sampling iteration. Figure 2 shows the execution time of 5,000 iterations of this step (3,000 burn-in and 2,000 sampling iterations, according to Gelfand et al., 2003), yielding a lower bound on the overall execution time. An experimental run with Bayesian inference for nonisotropic GP priors requires 230 CPU core days even for 100 training instances; as matrix inversion scales nearly cubically in n, it is impractical for this application. We therefore exclude this method from the remaining experiments. By contrast, full Bayesian inference in our GP model takes less than a second. The execution time of the kernel-local smoothing varying-coefficient model by Fan and Zhang (2008) substantially differs for the regression and classification task. In this model, separate point estimates of model parameters have to be inferred for each test instance, for which a separate optimization problem needs to be solved. For regression, efficient closed-form solutions for parameter estimates are available, while for classification more expensive numerical optimization is required (Fan and Zhang, 2008).

In all subsequent experiments, each method is given 30 CPU core days of execution time; experiments are run sequentially for increasing number n of training instances and results are reported for values of n for which the cumulative execution time is below this limit.

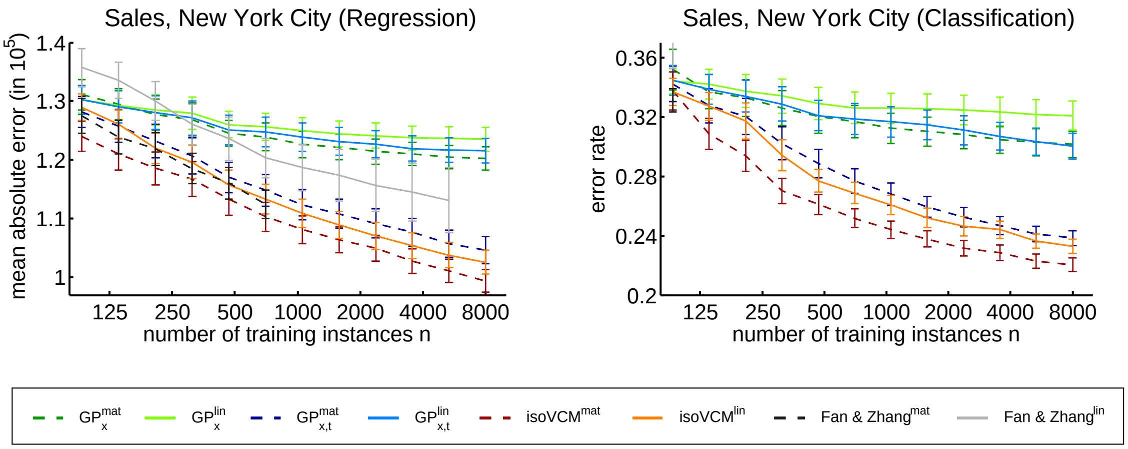

Figure 3: Mean absolute error for predicting real estate prices in New York City (left) and mean zero-one loss for classifying real estate transactions (right) over training set size n. Error bars indicate the standard error.

Figure 3 shows the mean absolute error for real estate price predictions (left) and the mean zero-one loss for classifying sales transactions (right) as a function of training set size n. For regression, *Fan & Zhang*^{lin} and *Fan & Zhang*^{mat} partially completed the experiments; for classification, both methods did not complete the experiment for the smallest value of n. All other methods completed the experiments within the time limit. For regression, we observe that $isoVCM^{lin}$ is substantially more accurate than $GP^{lin}_{\mathbf{x}}$, $GP^{lin}_{\mathbf{x},\mathbf{t}}$, and *Fan & Zhang*^{lin}; $isoVCM^{mat}$ is more accurate than $GP^{mat}_{\mathbf{x}}$ and $GP^{mat}_{\mathbf{x},\mathbf{t}}$ with $p < 0.01$ for all training set sizes according to a paired t-test. Significance values of paired t-test comparing $isoVCM^{mat}$ and *Fan & Zhang*^{mat} fluctuate between $p < 0.01$ and $p < 0.2$ for different n, indicating that $isoVCM^{mat}$ is likely more accurate than *Fan & Zhang*^{mat}. For classification, $isoVCM^{lin}$ substantially outperforms $GP^{lin}_{\mathbf{x}}$ and $GP^{lin}_{\mathbf{x},\mathbf{t}}$; $isoVCM^{mat}$ outperforms $GP^{mat}_{\mathbf{x}}$ and $GP^{mat}_{\mathbf{x},\mathbf{t}}$ ($p < 0.01$ for $n > 125$).

Figure 4 shows the mean absolute error for predicting monthly housing rent (left) and the mean zero-one loss for classifying rental contracts (right) for rental contracts in the state of California (upper row) and the state of New York (lower row) as a function of training set size n. *Fan & Zhang*^{lin} completed the regression experiments within the time limit and partially completed the classification experiment; *Fan & Zhang*^{mat} partially completed the regression experiment but did not complete the classification experiment for the smallest value of n. We again observe that $isoVCM^{mat}$ yields the most accurate predictions for both classification and regression problems; $isoVCM^{lin}$ always yields more accurate predictions than *Fan & Zhang*^{lin} and more accurate predictions than $GP^{lin}_{\mathbf{x},\mathbf{t}}$ for training set sizes larger than $n = 1000$.

5. Discussion and Related Work

Varying-coefficient models reflect applications in which a conditional distribution of y given $\mathbf{x}$ is a function of task variables $\mathbf{t}$. The task variables can, for instance, be continuous, discrete, or nodes in a tree—as in hierarchical Bayesian multitask learning. The functional dependency

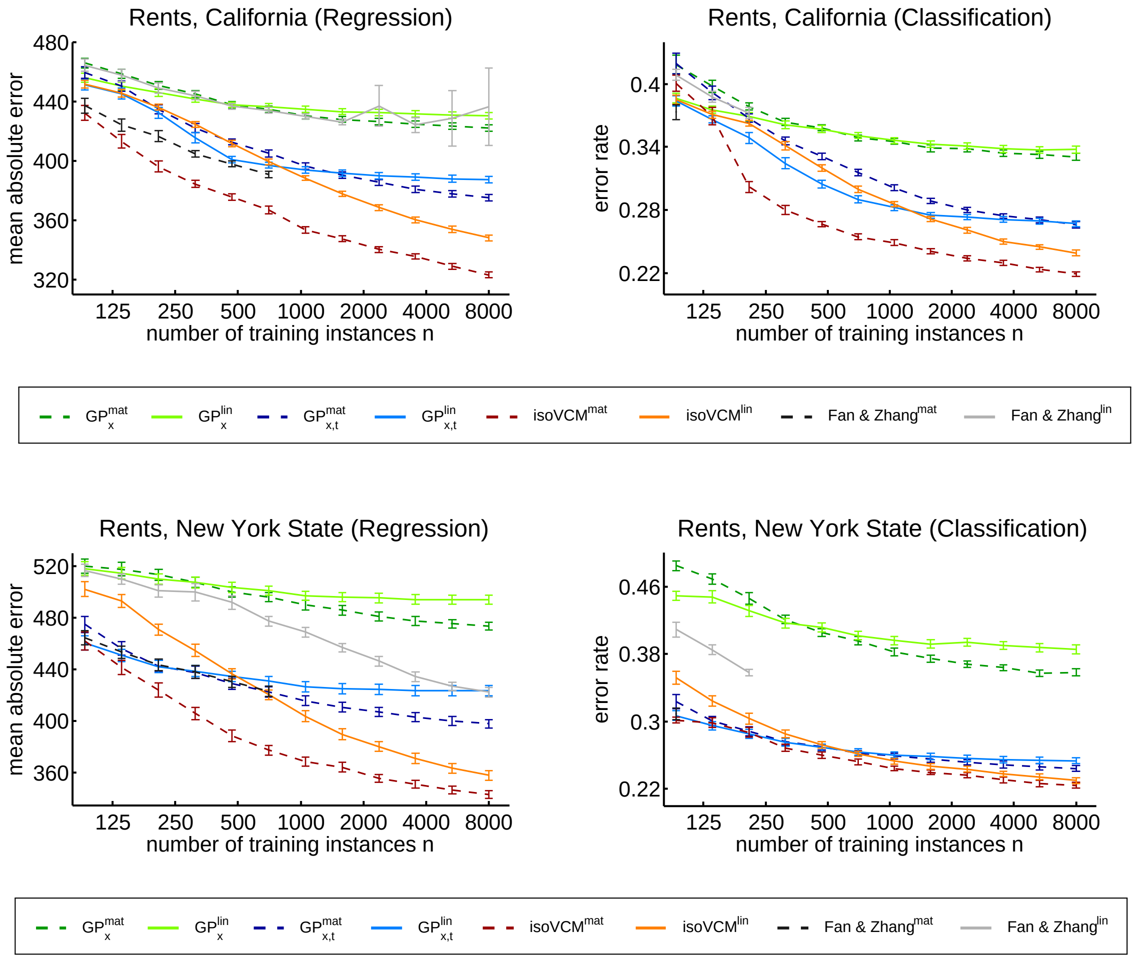

Figure 4: Mean absolute error for predicting monthly housing rents (left) and mean zero-one loss for classifying rental contracts (right) in the states of California (upper row) and New York (lower row) over training set size n. Error bars indicate the standard error.

between the conditional distribution of the output given the input and the task variables can be modeled with a GP prior. Theorem 3 shows that, for isotropic GP priors, Bayesian inference in varying-coefficient models can be carried out efficiently by using a standard Gaussian process with a kernel that is defined as the product of a task kernel and an instance kernel. This result clarifies the exact modeling assumptions required to derive the multitask kernel of Bonilla et al. (2007). This result also highlights that Bayesian inference for hierarchical Bayesian learning can be carried out efficiently by using a standard Gaussian process with graph-Laplacian kernel (Evgeniou et al., 2005).

Product kernels play a role in other multitask learning models. In the linear coregionalization model, several related functions are modeled as linear combinations of Gaussian processes; the covariance function then resolves to a product of a kernel function on instances and a matrix of mixing coefficients (Journel and Huijbregts, 1978; Álvarez et al., 2011). A similar model is studied by Wang et al. (2007) in the context of style-content separation in human locomotion data; here mixing coefficients are given by latent variables that represent an individual's movement style. Zhang and Yeung (2010) study a model for learning task relationships, and show that under a matrix-normal regularizer the solution of a multitask-regularized risk minimization problem can be expressed using a product kernel. Theorem 3 can be seen as a generalization of their result in which the regularizer is replaced by a prior over functions, and the regularized risk minimization perspective by a fully Bayesian analysis.

Non-stationarity can also be modeled in Gaussian processes by assuming that either the residual variance (Wang and Neal, 2012), or the length scale of the covariance function (Schmidt and O' Hagan, 2003), or the amplitude of the output (Adams and Stegle, 2008) are input-dependent. The varying-coefficient model differs from these models in that the source of non-stationarity is observed in the task variable.

In the domain of real estate price prediction, the dependency between property attributes and the market price changes continuously with geographical coordinates and time. We observe that primal Bayesian inference in varying-coefficient models with nonisotropic GP priors is all but impractical in this domain, while for isotropic GP priors, inference based on Theorem 3 is more efficient by several orders of magnitude. Empirically, we observe that the linear and kernelized *isoVCM* models predict real estate prices and housing rents more accurately over time and space than kernel-local smoothing varying-coefficient models, and are also more accurate than linear and kernelized models that append the task variables to the attribute vector or ignore the task variables.

Acknowledgments

We would like to thank Jörn Malich and Ahmed Abdelwahab for their help in preparing the data sets of monthly housing rents. We gratefully acknowledge support from the German Research Foundation (DFG), grant LA 3270/1-1.

Appendix

Proof of Proposition 2.

The marginal $p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \boldsymbol{\kappa})$ is defined by the generative process of drawing $\boldsymbol{\omega} \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\kappa})$, evaluating $\boldsymbol{\omega}$ for the k different tasks to create parameter vectors $\boldsymbol{\omega}(1), \ldots, \boldsymbol{\omega}(k)$, and then drawing $y_i \sim p(y|\mathbf{x}_i, \boldsymbol{\omega}(\mathbf{t}_i))$ for $i = 1, \ldots, n$. The marginal $p(\mathbf{y}|\mathbf{X}, \mathbf{T}; \mathcal{G}, \boldsymbol{\sigma})$ is defined by the generative process of generating parameter vectors $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_k$ according to Equations 3 and 4 in Definition 1, and then drawing $y_i \sim p(y|\mathbf{x}_i, \bar{\mathbf{w}}_{\mathbf{t}_i})$ for $i = 1, \ldots, n$. Here, the observation models $p(y|\mathbf{x}_i, \bar{\mathbf{w}}_{\mathbf{t}_i})$ and $p(y|\mathbf{x}_i, \boldsymbol{\omega}(\mathbf{t}_i))$ are identical. It therefore suffices to show that $p(\boldsymbol{\omega}(1), \ldots, \boldsymbol{\omega}(k)|\boldsymbol{\kappa}) = p(\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_k|\mathcal{G}, \boldsymbol{\sigma})$.

The distribution $p(\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_k|\mathcal{G}, \boldsymbol{\sigma})$ can be derived from standard results for Gaussian graphical models. Let $\bar{\mathbf{W}} \in \mathbb{R}^{k \times m}$ denote the matrix with row vectors $\bar{\mathbf{w}}_1^\mathsf{T}, \ldots, \bar{\mathbf{w}}_k^\mathsf{T}$, and let $\mathrm{vec}(\bar{\mathbf{W}}^\mathsf{T}) \in \mathbb{R}^{km}$ denote the vector of random variables obtained by stacking the vectors $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_k$ on top of another. According to Equations 3 and 4, the distribution over the random variables within $\mathrm{vec}(\bar{\mathbf{W}}^\mathsf{T})$ is given by a Gaussian graphical model (*e.g.,* Murphy (2012), Chapter 10.2.5) with weight matrix $\mathbf{A} \otimes \mathbf{I}_{m \times m} \in \mathbb{R}^{km \times km}$ and standard deviations $\boldsymbol{\sigma} \otimes \mathbf{1}_m$, where $\mathbf{1}_m \in \mathbb{R}^m$ is the all-one vector. It follows that the distribution over $\mathrm{vec}(\bar{\mathbf{W}}^\top) \in \mathbb{R}^{km}$ is given by

$$p(\mathrm{vec}(\bar{\mathbf{W}}^\top)|\mathcal{G}, \boldsymbol{\sigma}) = \mathcal{N}(\mathrm{vec}(\bar{\mathbf{W}}^\top)|\mathbf{0}, \bar{\boldsymbol{\Sigma}})$$

with

$$\begin{aligned}\bar{\boldsymbol{\Sigma}} = (\mathbf{I}_{km\times km} - \mathbf{A} \otimes \mathbf{I}_{m\times m})^{-1} \, &\mathrm{diag}(\boldsymbol{\sigma} \otimes \mathbf{1}_m)^2 \\ &(\mathbf{I}_{km\times km} - \mathbf{A}^\top \otimes \mathbf{I}_{m\times m})^{-1}\end{aligned}$$

(see Murphy (2012), Chapter 10.2.5), where $\mathrm{diag}(\boldsymbol{\sigma} \otimes \mathbf{1}_m) \in \mathbb{R}^{km\times km}$ denotes the diagonal matrix with entries $\boldsymbol{\sigma} \otimes \mathbf{1}_m$.

The distribution $p(\boldsymbol{\omega}(1), \ldots, \boldsymbol{\omega}(k)|\boldsymbol{\kappa})$ is given directly by the Gaussian process defining the prior over vector-valued functions $\boldsymbol{\omega} : \mathcal{T} \to \mathbb{R}^m$ (see Equation 1). Let $\boldsymbol{\Omega} \in \mathbb{R}^{k\times m}$ denote the matrix with row vectors $\boldsymbol{\omega}(1)^\top, \ldots, \boldsymbol{\omega}(k)^\top$, then the Gaussian process prior implies

$$p(\mathrm{vec}(\boldsymbol{\Omega}^\top)|\boldsymbol{\kappa}) = \mathcal{N}(\mathrm{vec}(\boldsymbol{\Omega})^\top|\mathbf{0}, \mathbf{G} \otimes \mathbf{I}_{m\times m})$$

(see, *e.g.,* Álvarez et al. (2011), Section 3.3). A straightforward calculation now shows $\mathbf{G} \otimes \mathbf{I}_{m\times m} = \bar{\boldsymbol{\Sigma}}$ and thereby proves the claim. ■

Proof of Proposition 6. In the following we use the notation that is introduced in Proposition 2 and Definition 5. We first observe that by the definition of the graph Laplacian multitask kernel it is sufficient to show that $\mathbf{G} = \mathbf{L}^\dagger$. Since the matrix $\mathbf{G}$ is invertible, this is equivalent to $\mathbf{G}^{-1} = \mathbf{L}$.

We prove the claim by induction over the number of nodes $|\mathcal{T}|$ in the tree $\mathcal{G}$. If $|\mathcal{T}| = 1$, then we have $\mathbf{A} = 0$, $\mathbf{D} = 0$, $\mathbf{R} = \sigma_1^{-2}$ and $\mathbf{M} = 0$. This leads to

$$\mathbf{G}^{-1} = (\mathbf{I} - \mathbf{A}^\top)\sigma_1^{-2}(\mathbf{I} - \mathbf{A}) = \sigma_1^{-1} = \mathbf{D} + \mathbf{R} - \mathbf{M} = \mathbf{L}$$

and proves the base case. Let us now assume that we have a tree $\mathcal{G}_k$ with $|\mathcal{T}| = k > 1$ nodes. Let $\mathbf{t}$ be a leaf of this tree and $\mathbf{t}'$ shall be its unique parent. Suppose we have $\mathbf{t}' = i$ and w.l.o.g. we assume that $\mathbf{t} = k$. Let furthermore $\mathcal{G}_{k-1}$ be the tree which we get by removing the node k and its adjacent edge from the tree $\mathcal{G}_k$. Let $\mathbf{A}_k$ and $\mathbf{A}_{k-1}$ denote the adjacency matrices and $\mathbf{D}_k$ and $\mathbf{D}_{k-1}$ the degree matrices of $\mathcal{G}_k$ and $\mathcal{G}_{k-1}$. Let $\boldsymbol{\sigma}_k \in \mathbb{R}^k$ be the vector with entries $\sigma_1, \ldots, \sigma_k$, and $\boldsymbol{\sigma}_{k-1} \in \mathbb{R}^{k-1}$ be the vector with entries $\sigma_1, \ldots, \sigma_{k-1}$. Let $\mathbf{R}_k \in \mathbb{R}^{k\times k}$ denote the diagonal matrix with entries $\sigma_1^{-2}, 0, \ldots, 0$, and $\mathbf{R}_{k-1} \in \mathbb{R}^{k-1\times k-1}$ the diagonal matrix with entries $\sigma_1^{-2}, 0, \ldots, 0$. Let $\mathbf{B}_k \in \mathbb{R}^{k\times k}$ denote the diagonal matrix with entries $0, \sigma_2^{-2}, \ldots, \sigma_k^{-2}$ and $\mathbf{B}_{k-1} \in \mathbb{R}^{k-1\times k-1}$ the diagonal matrix with entries $0, \sigma_2^{-2}, \ldots, \sigma_{k-1}^{-2}$. Let $\mathbf{M}_k = \mathbf{B}_k\mathbf{A}_k + (\mathbf{B}_k\mathbf{A}_k)^\top$ and $\mathbf{M}_{k-1} = \mathbf{B}_{k-1}\mathbf{A}_{k-1} + (\mathbf{B}_{k-1}\mathbf{A}_{k-1})^\top$. Let $\mathbf{L}_k = \mathbf{D}_k + \mathbf{R}_k - \mathbf{M}_k$ and $\mathbf{L}_{k-1} = \mathbf{D}_{k-1} + \mathbf{R}_{k-1} - \mathbf{M}_{k-1}$.

In the following, we write $\mathrm{diag}(\mathbf{v})$ to denote a diagonal matrix with entries $\mathbf{v}$. We then have

$$\mathbf{A}_k = \left(\begin{array}{c|c} \mathbf{A}_{k-1} & \mathbf{e} \\ \hline \mathbf{0} & 0 \end{array}\right), \text{ where } \mathbf{e} = (\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0)^\top$$

is the i^{th} $(n-1)$-dimensional unit vector. Using this notation we can write

$$\begin{aligned}
\mathbf{G}_k^{-1} &= (\mathbf{I} - \mathbf{A}_k^{\mathsf{T}})\,\mathrm{diag}(\boldsymbol{\sigma}_k)^{-2}(\mathbf{I} - \mathbf{A}_k) \\
&= \left(\begin{array}{c|c} \mathbf{I} - \mathbf{A}_{k-1}^{\mathsf{T}} & \mathbf{0} \\ \hline -\mathbf{e}^{\mathsf{T}} & 1 \end{array}\right) \left(\begin{array}{c|c} \mathrm{diag}(\boldsymbol{\sigma}_{k-1})^{-2} & \mathbf{0} \\ \hline \mathbf{0} & \sigma_k^{-2} \end{array}\right) \left(\begin{array}{c|c} \mathbf{I} - \mathbf{A}_{k-1} & -\mathbf{e} \\ \hline \mathbf{0} & 1 \end{array}\right) \\
&= \left(\begin{array}{c|c} \mathbf{L}_{k-1} + \sigma_k^{-2}\mathbf{e}\mathbf{e}^{\mathsf{T}} & -\sigma_k^{-2}\mathbf{e} \\ \hline -\sigma_k^{-2}\mathbf{e}^{\mathsf{T}} & \sigma_k^{-2} \end{array}\right).
\end{aligned}$$

In the last line we applied the induction hypothesis to the tree $\mathcal{G}_{k-1}$. Using the definitions of $\mathbf{L}$, $\mathbf{D}$, $\mathbf{R}$ and $\mathbf{M}$, we can easily finish the proof:

$$\begin{aligned}
\mathbf{G}_k^{-1} &= \left(\begin{array}{c|c} \mathbf{D}_{k-1} + \mathbf{R}_{k-1} - \mathbf{M}_{k-1} + \sigma_k^{-2}\mathbf{e}\mathbf{e}^{\mathsf{T}} & -\sigma_k^{-2}\mathbf{e} \\ \hline -\sigma_k^{-2}\mathbf{e}^{\mathsf{T}} & \sigma_k^{-2} \end{array}\right) \\
&= \left(\begin{array}{c|c} \mathbf{D}_{k-1} + \sigma_k^{-2}\mathbf{e}\mathbf{e}^{\mathsf{T}} & \mathbf{0} \\ \hline \mathbf{0} & \sigma_k^{-2} \end{array}\right) + \left(\begin{array}{c|c} \mathbf{R}_{k-1} & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right) - \left(\begin{array}{c|c} \mathbf{M}_{k-1} & \sigma_k^{-2}\mathbf{e} \\ \hline \sigma_k^{-2}\mathbf{e}^{\mathsf{T}} & 0 \end{array}\right) \\
&= \mathbf{D}_k + \mathbf{R}_k - \mathbf{M}_k \\
&= \mathbf{L}_k.
\end{aligned}$$

This proves the claim. ■

References

R. P. Adams and O. Stegle. Gaussian process product models for nonparametric nonstationarity. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. Technical Report MIT-CSAIL-TR-2011-033, Massachusetts Institute of Technology, 2011.

E. V. Bonilla, F. V. Agakov, and C. K. I. Williams. Kernel multi-task learning using task-specific features. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615–637, 2005.

J. Fan and T. Huang. Profile likelihood inferences on semiparametric varying-coefficient partially linear models. *Bernoulli*, 11(6):1031–1057, 2005.

J. Fan and W. Zhang. Statistical methods with varying coefficient models. *Statistics and Its Interface*, 1(1):179–195, 2008.

J. R. Finkel and C. D. Manning. Hierarchical Bayesian domain adaptation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

A. E. Gelfand, H. Kim, C. F. Sirmans, and S. Banerjee. Spatial modeling with spatially varying coefficient processes. *Journal of the American Statistical Association*, 98(462):387–396, 2003.

A. Gelman, J. B. Carlin, H. S. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

T. Hastie and R. Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society*, 55 (4):757–796, 1993.

A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978.

Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, MA, 2012.

C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

A. M. Schmidt and A. O' Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society*, 65(3):745–758, 2003.

B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT Press, 2002.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, 2005.

C. Wang and R. M. Neal. Gaussian process regression with heteroscedastic or non-Gaussian residuals. Technical Report CoRR abs/1212.6246, University of Toronto, 2012.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Multifactor Gaussian process models for style-content separation. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

D. C. Wheeler and C. A. Calder. Bayesian spatially varying coefficient models in the presence of collinearity. *American Statistical Association, Spatial Modeling Section*, 2006.

C. O. Wu and C. T. Chiang. Kernel smoothing on varying coefficient models with longitudinal dependent variable. *Statistica Sinica*, 10(1):433–456, 2000.

R. Yan and J. Zhang. Transfer learning using task-level features with application to information retrieval. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

Y. Zhang and D. Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.